

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Ms. Toner (reg. 52,142) on 8/13/2009.

The application has been amended as follows:

In the claims:

1. (Currently Amended) A method for detecting software development best practice violations, comprising:

receiving sets of source code from a plurality of sources;

extracting at least one code pattern from the sets of source code;

defining meta data for each of the at least one code pattern that indicates a quality of the at least one code pattern;

classifying and assigning a rank to each of the at least one code pattern based on the corresponding meta data, a skill level and an experience level of a developer of the at least one code pattern;

storing each of the at least one code pattern and the assigned rank in a data structure;

receiving a subsequent set of source code having no affiliation with, and developed independently from all of the at least one code pattern stored in the data structure;

extracting and classifying a code pattern to be tested from the subsequent set of source code;

comparing the code pattern to be tested to the at least one code pattern classified in a same category stored in the data structure to determine a closest match to the code pattern to be tested;

assigning a rank of the closest match to the code pattern to be tested; and

detecting a software development best practice violation if the rank assigned to the code pattern to be tested fails to comply with a predetermined threshold.

2. (Canceled)

10. (Currently Amended) A method for building a dynamic best practice violation (BPV) engine resource for detecting software development best practice violations, comprising:

receiving sets of source code from a plurality of sources;

detecting a programming language of each of the sets of source code;

extracting a plurality of code patterns from the sets of source code;

defining meta data for each of the plurality of code patterns that indicates a quality of the plurality of code patterns;

classifying and assigning a rank to each of the plurality of code patterns based on the corresponding meta data, a skill level and an experience level of a developer of each of the sets of source code;

storing each of the plurality of data patterns and the assigned rank in a data structure;

receiving a subsequent set of source code having no affiliation with the plurality of code patterns stored in the data structure;

extracting and classifying a code pattern to be tested from the subsequent set of source code plurality of code patterns in the data structure developed independently from the plurality of code patterns stored in the data structure;

comparing the code pattern to be tested to the plurality of code patterns classified in a same category stored in the data structure to determine a closest match to the code pattern to be tested;

assigning a rank of the closest match to the code pattern to be tested; and

detecting a software development best practice violation if the rank assigned to the code pattern to be tested fails to comply with a predetermined threshold.

15. (Canceled)

18. (Currently Amended) A method for detecting software development best practice violations, comprising:

receiving sets of source code from a plurality of sources;

detecting a programming language of each of the sets of source code;

extracting [[the]] a plurality of code patterns from the sets of source code;

defining meta data for each of the plurality of code patterns that indicates a quality of the plurality of code patterns; and

classifying and assigning a rank to each of the plurality of code patterns based on the corresponding meta data, a skill level and an experience level of a developer of each of the other sets of source code:

storing the plurality of code patterns and the ranks in a best practice violation (BPV) engine resource,

receiving a first set of source code in the BPV engine, the first set of source code having no affiliation with, and developed independently from the plurality of code patterns;

extracting and classifying a code pattern to be tested from the first set of source code;

comparing the code pattern to be tested to [[a]] the plurality of code patterns classified in a same category extracted from the sets of source code previously received and analyzed by the BPV engine to determine a closest match to the code pattern to be tested;

assigning a rank previously assigned to the closest match to the code pattern to be tested;

and

detecting a software development best practice violation if the rank assigned to the code pattern to be tested fails to comply with a predetermined threshold.

20. (Canceled).

In the abstract:

In line 5 and 7, “thereof” has been deleted.

In line 6, “it’s” has been changed to –its--.

Title

The title has been changed to “Method, system and program product for detecting deviation from software development best practice resource in a code sharing system.”

Examiner’s Statement of Reason(s) for Allowance

2. Claims 1, 3-8, 10-14, 16, 18, and 21-24 (renumbered as 1-18) are allowed.

3. The following is an examiner’s statement of reason s for allowance:

The closest prior arts of record, taken alone or in combination, fail to teach or fairly suggest at least: classifying and assigning a rank to each of the plurality of code patterns based on the corresponding meta data, a skill level and an experience level of a developer of each of the other sets of source code...receiving a first set of source code in the BPV engine, the first set of source code having no affiliation with, and developed independently from the plurality of code patterns...detecting a software development best practice violation if the rank assigned to

the code pattern to be tested fails to comply with a predetermined threshold” as recited in the independent claims.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled “Comments on Statement of Reasons for Allowance.”

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to INSUN KANG whose telephone number is (571)272-3724. The examiner can normally be reached on M-R 7:30-6 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Lewis A. Bullock, Jr. can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Insun Kang/
Primary Examiner, Art Unit 2193